Polysense WxS8000 User Guide

Mar 26, 2018
Document Number: UG-XXXX
Revision 2.3

Table of Contents

1. OVERVIEW 5

2. WXS8000 INTRODUCTION 6

2.1. ANTENNA 7
2.2. POWER SOURCE 7

3. QUICK START 8

4. WXS8000 INTERFACE WITH LORA SERVER 10

4.1. DOWNLINK MESSAGE 10
4.1.1. SET INDIVIDUAL ATTRIBUTE IN PPB (PLSSSETPPB) 11
4.1.1.1. GENERAL ATTRIBUTES 11
4.1.1.2. SENSOR SPECIFIC ATTRIBUTES 14
4.1.2. SET INDIVIDUAL ATTRIBUTE IN PRB (PLSSSETPRB) 18
4.1.3. SET MOTE LOCAL DATE-TIME (PLSSSETCLK) 18
4.1.4. SET COUNTER SENSOR (PLSSSETEXTCNTR) 19
4.1.5. GET INDIVIDUAL ATTRIBUTE IN PPB (PLSSGETPPB) 19
4.2. UPLINK MESSAGE 19
4.2.1. SENSOR DATA REPORT WITH FIXED DATA LENGTH 20
4.2.2. SENSOR DATA REPORT WITH VARIABLE DATA LENGTH 23
4.2.3. MOTE INFORMATION REPORT 25
4.2.4. MOTE LOCAL DATE-TIME REPORT 26
4.2.5. MOTE STATISTICS REPORT 26
4.2.6. PPB ATTRIBUTE REPORT 26

5. FIRMWARE UPGRADE 28

5.1. FIRMWARE UPGRADE VIA ST-LINK 28
5.1.1. SETUP ST-LINK AND ST-LINK UTILITY 28
5.1.2. FIRMWARE UPGRADE 31
5.1.2.1. UPGRADE OLD WXS8000 FIRMWARE – SINGLE IMAGE 31
5.1.2.2. UPGRADE NEW WXS8000 FIRMWARE – LOADER + APPLICATION IMAGES 32
5.2. FIRMWARE UPGRADE VIA OTA 33
5.2.1. PREPARE FW IMAGE 33
5.2.2. UPGRADE FW 33

6. COMPLIANCES 35

CONTACT INFORMATION 36

Revision History

Revision	Date	History
1.0	10/26/2017	First revision
1.1	11/09/2017	Added Chapter 4
1.2	11/15/2017	. Corrected the wrong FPort in downlink frame, should be 251 instead of 254 . Corrected the wrong attribute# in 4.1.1.2
1.3	11/22/2017	. Added Chapter 5
1.4	12/14/2017	. Added external ultrasonic sensor type (0xA) and CO2 (0x29) sensor type . Added Distance and CO2 concentration in % data types . Added more regions
1.5	12/28/2017	. Added new external CO2 sensor support for environment air CO2 quality monitoring as well as new CO2 concentration in ppm data type . Supported AS923 region . Supported Carrier sensing mechanism and configuration in PPB . Supported cross-threshold data report according to the pre-configured 2 thresholds in PPB . Supported new PlssGetPPB message (refer to section 4.1.5), and PPB attribute report message (refer to section 4.2.4) . Updated the endian rule in both uplink and downlink message (refer to section 4.1 and 4.2) . The downlink message type can be confirmed or unconfirmed (refer to section 4.1)
1.6	01/09/2018	. Added new external current sensor and external pressure sensor as well as new data types in data report . Support periodic data report in parallel with cross-threshold data report
1.7	01/15/2018	. Added direction connection mappings between ST-Link JTAG pins and WxS8000 JTAG pins in section 5.1 . Added ST-Link settings
1.8	01/18/2018	. Supported proximity sensor . Added an example for proximity sensor settings
1.9	01/24/2018	. Added the channel planning settings on gateway (refer to chapter 3) . Updated sensor DATA_THRESHOLD1 comment
2.0	02/09/2018	. Added firmware upgrade via OTA

2.1	03/06/2018	. Added Air quality/PM monitor sensor
2.2	03/21/2018	. Added section 4.2.2 Sensor Data Report with Variable Data Length
2.3	03/26/2018	. Added Compliances chapter

1. Overview

Polysense WxS8000 is a powerful product portfolio which can support a very wide range of IoT sensing applications, including but not limited to, smart/safe building, smart/safe city, perimeter security, fire and gas monitoring, precision agriculture, object tracking, machine condition monitoring, smart metering, utility pole tilt monitoring.

WxS8000 supports the LPWAN/LoRa as the uplink interface to transmit the sensor data to the application server via the LoRa network.

WxS8000 supports below key functions:

1. Support outdoor deployment with IP64+ enclosure IP rating
2. Support 1 or 2 AA battery or external DC 12V power supply
3. Support internal coil antenna or external SMA antenna for LoRa channel plans
4. Support NA, Europe and China ISM frequency bands (902.3~914.9MHz for US 902-928MHz ISM Band, 863.1~869.9 MHz for EU 863-870MHz ISM Band, 470.3~489.3MHz for CN 470-510MHz Band)
5. Support up to 30 sensor types, refer to section 4.1.1.2 for the sensor types

This document provides the guide to use WxS8000.

2. WxS8000 Introduction

Diagram 2-1 shows the appearance of WxS8000

Diagram 2-1 WxS8000 Appearance

Diagram 2-2 shows the internal looking of WxS8000

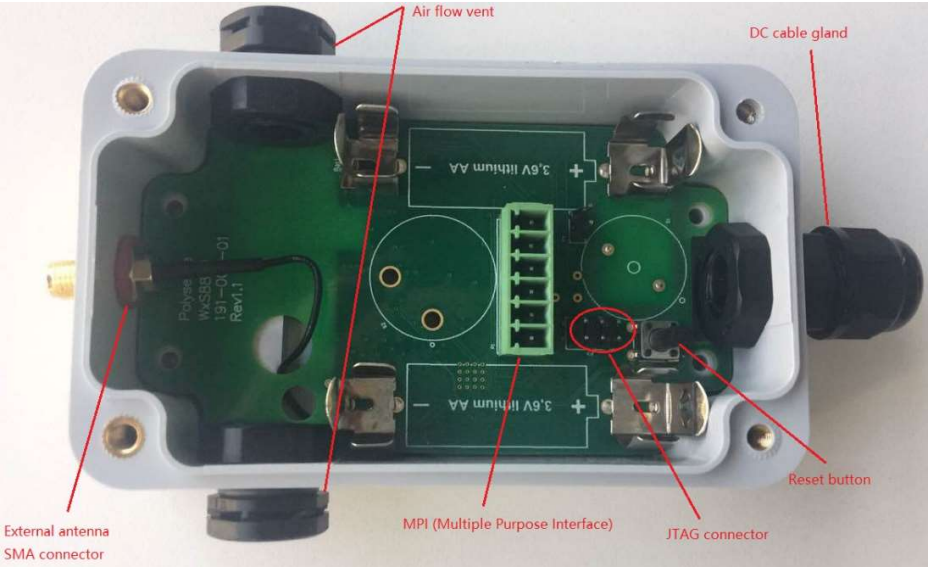


Diagram 2-2 WxS8000 Internal Looking

NOTE:

1. The 2 air flow holes/vents are optional, if there are no install internal temperature/humidity/CO/sound/pressure sensors installed, then there are no air flow holes on the enclosure

2. The DC cable hole/gland is also optional, if WxS8000 is powered by AA battery, and there is also no external sensor to be connected to WxS8000, then there is no DC cable hole on the enclosure

2.1. Antenna

WxS8000 supports internal coil antenna and external antenna, by default, it will use external antenna, there is a SMA tail cable installed to connect the IPX socket on PCB to external antenna, user just needs to screw the external antenna to the SMA connector on the enclosure.

2.2. Power source

If AA battery is used as power source, it is strongly recommended to use the 3.6V LI/SOCl2 AA battery. User can use one AA battery, or 2 AA batteries which can double the life cycle of the battery power supply.

Please follow the battery polarity mark on PCB to ensure the battery is installed correctly.

Battery power supply and external DC power supply are mutually exclusive; they can't be used to power WxS8000 at the same time.

3. Quick Start

By default, WxS8000 works in OTA (Over The Air) activation mode, below table shows WxS8000 default settings, which are saved in WxS8000 local FLASH and called as PPB (Polysense Personality Base). For individual WxS8000 specific settings, including MoteEUI, sensor specific configurations, please refer to Polysense WxS8000 Factory Settings document.

User can follow below procedure to connect WxS8000 to LoRa network:

1. Configure the LoRa server to add the new end-device (or mote) according to WxS8000 default settings
2. If US902 or CN470 region is being used, then it is necessary to follow below settings to configure the LoRa gateway upstream channel plan (in order to align with WxS8000 factory settings):
   - US902: 902.3, 902.5, 902.7, 902.9, 903.1, 903.3, 903.5, 903.7
   - CN470: 470.3, 470.5, 470.7, 470.9, 471.1, 471.3, 471.5, 471.7
3. Install the external antenna (screw the external antenna on the SMA connector on enclosure) if the WxS8000 doesn't have internal coil antenna installed
4. Power on WxS8000 either via connecting the DC power supply or plugging the 3.6V LI/SOCl2 AA battery into the battery holder (if the AA battery is used as power source, PLEASE *ensure* the AA battery polarities are consistent with the silk-screen on PCB)
   - NOTE: There is NO AA battery inside WxS8000 box initially
5. WxS8000 starts the OTA join procedure and try to register to the nearby LoRa gateway/server

Item	Value
Region	1 (CN)
Scheduling granularity	30 seconds
LoRa class	0 (Class A)
Join type	0 (OTA - Over The Air)
Disable PRB (Polysense Runtime Base)	0 (disabled, so that WxS8000 will always initiate OTA procedure every time after reboot)
Enable ADR	1 (enabled)
Use fixed device nonce	0 (WxS8000 will use the device nonce generated randomly when joining the server in OTA mode)
Device EUI	(Refer to the label on WxS8000 enclosure)
Application EUI	0
Application key	11-22-33-44-55-66-77-88-99-00-AA-BB-CC-DD-EE-FF

Sensor type 0	0 (Battery Voltage)
Presence 0	1 (the sensor is present on the board)
Administration state 0	1 (the sensor is enabled)
Scheduling cycle 0	120 (so the sensor scheduling cycle is 30 * 120 = 3600 seconds)
Sample number 0	1 (sample sensor data only for once in each scheduling cycle)
Sample cycle 0	N/A
<Sensor type n>	Sensor type n specific configuration if applicable

Table 3.1-1 WxS8000 PPB default settings

NOTE:

1. If the sample number is set to >= 2, then it will enable the dense data sample for this sensor type (i.e. the sensor type will be sampled for multiple times as specified by the sample number in one scheduling cycle)
2. Only Vibration sensor types are supported with dense data sample at this point, including Vibration with 3-axis and Vibration with 2-axis
3. It only supports to do the dense data sample for one sensor type at one time, if Vibration with 3-axis and Vibration with 2-axis are both enabled with the dense data sample (both are set as sample number >=2), then only the Vibration with 2-axis sensor type will be enabled actually

4. WxS8000 Interface with LoRa Server

This chapter defines Polysense WxS8000 interface with LoRa server over the air, the interfaces include WxS8000 configurations initiated from LoRa server, sensor data report initiated from WxS8000. The configurations or sensor data report is named as Polysense proprietary messages which is filled into the FRMPayload in LoRa frame as shown below:

Size (bytes)	7..22	0..1	0..*N*
MACPayload	FHDR	FPort	FRMPayload

- The sensor data in FRMPayload is in binary format

4.1. Downlink Message

The downlink messages initiated from LoRa server are used to configure WxS8000, including WxS8000 LoRa mote specific configurations, WxS8000 sensor configurations, etc. All the downlink messages are unconfirmed even though some messages request WxS8000 to give response.

- The message type (MType) in MAC header can be 0x03 (Unconfirmed Data Down) or 0x05 (Confirmed Data Down) depending on LoRa server demand
- The FPort in the MACPayload is 251 (reserved FPort in LoRa spec)
- For the field in FRMPayload, if its length is 2-byte or 4-byte, then it is in little endian, if its length is 8-byte (APP_EUI and DEVICE_EUI), then it is in big endian. For example,
  - For SCHEDULING_CYCLE whose length is 2-byte, if user wants to configure it as 0x0A, then in FRMPayload, it must be set as 0A00
  - For APP_EUI whose length is 8-byte, if user wants to configure it as 0x11223344, then in FRMPayload, it must be set as 0000000011223344 (the prefixed zeroes are necessary)
- The basic formats of downlink message:
  - *Format#1*

Size (bytes)	1
FRMPayload	CID

  - *Format#2*

Size (bytes)	1	1/4/6/200
FRMPayload	CID	Value

  - *Format#3*

Size (bytes)	1	1	1	0..N
FRMPayload	CID	Attribute	ValueLength	Value

- CID: command ID to identify one specific command initiated from LoRa server, below table shows all the commands:

CID	Command	Format#	Comment
0x2	PlssReboot	1	Reboot WxS8000 immediately
0x8	PlssGetInfo	1	Get WxS8000 management information, WxS8000 should respond this with Mote information report message in next uplink transmission window
0x13	PlssSetPPB	3	Set specific attribute in WxS8000 PPB (Polysense Proprietary Base)
0x14	PlssSetPRB	3	Set specific attribute in WxS8000 PRB (Polysense Runtime Base)
0x15	PlssSetClk	2	Set WxS8000 local date-time to align with the server
0x19	PlssSetExtCntr	2	Set the counter of the external counter sensor to the value specified in Value field
0x1A	PlssGetPPB	3	Get specific attribute in WxS8000 PPB, and there is NO Value parameter in format#3 above

- Value: the value to be set to WxS8000
- Attribute: identify one specific attribute in PPB or PRB
- ValueLength: the length of the value in octet

- If the FRMPayload length allows (the MAX FRMPayload length is region-dependent), LoRa server can pack multiple downlink messages together and fill into the FRMPayload of one LoRa frame in order to improve the transmission efficiency. Once WxS8000 receives the frame, it will parse out all the messages, and process one by one.
- If there is PlssReboot message inside the FRMPayload which consists of multiple messages, then WxS8000 won't execute reboot until all other messages are handled

4.1.1. Set Individual Attribute in PPB (PlssSetPPB)

The change to PPB attributes can take effect immediately except the change of REGION, which will take effect only after WxS8000 reboot or power recycle.

4.1.1.1. General Attributes

Attribute	Description	ValueLength (bytes)	Value	Comment
0x2	REGION	1	0 – US902	LoRa region setting. WxS8000 supports all the regions as defined in LoRaWAN spec 1.0.2, besides WxS8000

			1 – CN470 2 – EU863 3 – CN779 4 – EU433 5 – EU470 6 – AS923	supports a special region – EU470. Note: the region setting in PPB must match the WxS8000 PCBA in order to support specific region.
0x4	PERIODIC_DATA_REPORT_CYCLE	2	Valid value range: 0x0~0xFFFF	PERIODIC_DATA_REPORT_CYCLE specifies periodic data report cycle in case cross-threshold data report is enabled, i.e. WxS8000 will report data periodically according to PERIODIC_DATA_REPORT_CYCLE in parallel with cross-threshold data report to ensure the server can receive the sensor data even the cross-threshold data report never occurs. The real periodic report cycle equals to SCHEDULING_GRANULARITY * SCHEDULING_CYCLE * PERIODIC_DATA_REPORT_CYCLE. If PERIODIC_DATA_REPORT_CYCLE is set <= 1, then the sampled data will always be reported regardless of crossing threshold or not (the cross-threshold check logic will be skipped in this case). Applicable if ENABLE_PERIODIC_DATA_REPORT is set to 1 and cross-threshold data report is enabled (i.e. the cross threshold report control bit in SENSOR_CTRL != 0 or DATA_THRESHOLD1 != 0). The cross-threshold data report event will restart the periodic data report timer, this can avoid additional periodic data report shortly after cross-threshold data report. As global setting, PERIODIC_DATA_REPORT_CYCLE is applicable to all sensors, i.e. the periodic data report cycle is same for all sensors, but the periodic data report feature can be disabled/enabled on per sensor base via ENABLE_PERIODIC_DATA_REPORT.
0x6	SCHEDULING_GRANULARITY	1	Valid value range: 0x3~0xFF	SCHEDULING_GRANULARITY specifies the system scheduling granularity (in unit of second), the sensor scheduling cycle equals to SCHEDULING_GRANULARITY * SCHEDULING_CYCLE. As global setting, the change to SCHEDULING_GRANULARITY will cause all sensors scheduling cycle to be changed.
0x7	JOIN_PRB_ADR	1	Bit[4] – Join type ▪ 0 – OTA (Over The Air) ▪ 1 – Personalization Bit[3] – Enable/disable PRB ▪ 0 – Disable PRB ▪ 1 – Enable PRB Bit[1] – Enable/disable ADR ▪ 0 – Disable ADR	PRB (Polysense Runtime Base) is designed to save WxS8000 important runtime session information, so that after WxS8000 is reboot, it can recover to a normal working state immediately by restoring the runtime session information from PRB, for example, in OTA mode, if PRB is enabled, WxS8000 will save the uplink counter, device address, application session key, network session key and channel group ID into PRB periodically, once WxS8000 is reboot, it will load such information from PRB and communicate with LoRa server directly without going through the join procedure, this will reduce the communication setup time between WxS8000 and LoRa server.

			▪ 1 – Enable ADR Bit[0] – Enable/disable Carrier Sensing ▪ 0 – Disable Carrier Sensing mechanism ▪ 1 – Enable Carrier Sensing mechanism **The unused bits MUST be set to 0**	
0x8	DEVICE_EUI	8		A global end-device ID in IEEE EUI64 address space that uniquely identifies one LoRa end-device **Applicable to OTA join-type only**
0x10	APP_EUI	8		A global end-device ID in IEEE EUI64 address space that uniquely identifies one LoRa application **Applicable to OTA join-type only**
0x18	APP_KEY	16		The APP_KEY is an AES-128 application key specific to one LoRa end-device **Applicable to OTA join-type only**
0x8	DEVICE_ADDR	4		DEVICE_ADDR identifies the LoRa end-device within the current LoRa network. It consists of NwkID[31..25] and NwkAddr[24..0]. **Applicable to Personalization join-type only**
0xC	APP_S_KEY	16		Application session key specific to one LoRa end-device **Applicable to Personalization join-type only**
0x1C	NWK_S_KEY	16		Network session key specific to one LoRa end-device **Applicable to Personalization join-type only**
0xBC	CHANNEL_GROUP	1	Bit[7~4] – the end group ID, valid data range: 0~0xF Bit[3~0] – the start group ID, valid data range: 0~0xF	The start/end channel group IDs specify the channel scope when WxS8000 is scanning to find the available channel after bootup, for example, if end group ID is 0, start group ID is 0, then WxS8000 will only scan channel#0 ~ #8 to communicate with gateway. NOTE: if PRB is enabled, WxS8000 will always use the channel group setting in PRB to communicate with LoRa gateway and server.
0xBD	PARTIME_SCHEDULING_FROM	2	Valid data range: 0~0x173B	Global start-time of part-time scheduling, below defines its format: 0xhhmm, for example, 0x173B refers as 23:59
0xBF	PARTIME_SCHEDULING_TO	2	Valid data range: 0~0x173B	Global end-time of part-time scheduling with the same format as PARTTIME_SCHEDULING_FROM It is possible that PARTTIME_SCHEDULING_TO is less than PARTTIME_SCHEDULING_FROM. In this case, PARTTIME_SCHEDULING_TO is the time of the second day. For example, if user wants to enable the sensor sampling in non-working-time (from 18:00 to 8:00 of the second day), he should set PARTTIME_SCHEDULING_FROM to 0x1200 (18:00), and set PARTTIME_SCHEDULING_TO to 0x0800 (8:00)

Some examples:

- The FRMPayload in hex to set scheduling granularity to 20 seconds: 13060114
- The FRMPayload in hex to set the scheduling cycle of second sensor to 1 (i.e. 1 x 20 = 20 seconds): 1340020100

  NOTE: the scheduling cycle is UINT16, as the message payload is little endian, so 0x0001 should be filled as 0x0100 in the payload

4.1.1.2. Sensor Specific Attributes

WxS8000 supports up to 8 sensors working simultaneously, below table shows how to configure the attributes of each sensor.

NOTE: N in the attribute column identified each individual sensor, its valid value is 0~7.

Attribute	Description	ValueLength (bytes)	Value	Comment
N*0x12 + 0x2C	SENSOR_TYPE	1	0 – Battery power 1 – Temperature 2 – Humidity 3 – Temperature + humidity 4 – Sound 5 – PIR 6 – Light 7 – Tilt 8 – Vibration with 3-axis 9 – GNSS (Global Navigation Satellite System) 0xA – External ultrasonic sensor 0xB – Atmosphere pressure 0xC – CO 0xD – Vibration with 2-axis 0xE – General external sensor with analog input 0xF – General external sensor with digital input 0x10 – External digital temperature sensor (DS18B20) 0x11 – External displacement sensor 0x12 – External soil humidity sensor 0x13 – External PIR sensor 0x14 – External counter sensor 0x15 – External switch sensor (including magnetic switch sensor: MC-56, and limit switch sensor: SZL-WL-K) 0x16 – External analog temperature sensor 0x17 – External NH3 sensor	Sensor types

			0x18 – AsH3 (Industry) 0x19 – C6H6 (Industry) 0x1A – Cl2 (Industry) 0x1B – H2 (Industry) 0x1C – H2S (Industry) 0x1D – HCL (Industry) 0x1E – HCN (Industry) 0x1F – HF (Industry) 0x20 – NH3 (Industry) 0x21 – NO2 (Industry) 0x22 – O3 (Industry) 0x23 – PH3 (Industry) 0x24 – SO2 (Industry) 0x25 – CH4 (MC112) 0x26 – C2H2 (MC119) 0x27 – Gasoline (MC119) 0x28 – C2H6O (MC119) 0x29 – CO2 (MD62) 0x2A – External CO2 (MH-Z19B) 0x2B – External current sensor 0x2C – External pressure sensor 0x2D – External proximity sensor 0x2E – Air quality/PM monitor sensor	
N*0x12 + 0x2D	SENSOR_CTRL	1	Bit[7] – Sensor presence ▪ 0 – The sensor isn't present ▪ 1 – The sensor is present Bit[6] – Sensor admin state ▪ 0 – Sensor is disabled ▪ 1 – Sensor is enabled Bit[3] – Periodic data report control (Refer to PERIODIC_DATA_REPORT_CYCLE for the usage of this setting) ▪ 0 – Disable periodic data report feature on this sensor ▪ 1 – Enable periodic data report on this sensor Bit[2] – Part-time scheduling ▪ 0 – Disable part-time sampling ▪ 1 – Enable part-time sampling Bit[1] – Cross threshold report control ▪ 0 – Enable the comparison against threshold with	If the sensor is not present (sensor presence bit is set to 0) or the sensor is disabled, the sensor will be ignored (neither be sampled nor reported with data). So user MUST set the sensor presence bit and sensor admin state bit to 1 in order to make the sensor work normally. If the part-time scheduling bit is set, the sensor will only work within the time interval from PARTTIME_SCHEDULING_FROM to PARTIME_SCHEDULEING_TO in a day, otherwise the sensor will work around the clock.

			the ABS value of the sampled sensor data, in this case only DATA_THRESHOLD2 will be used, it is an unsigned number ▪ 1 – Enable the comparison against the threshold with sampled sensor data directly, in this case, both DATA_THRESHOLD1 and DATA_THRESHOLD2 are used, and they are the signed numbers **The unused bits MUST be set to 0**	
N*0x12 + 0x2E	SCHEDULING_CYCLE	2	Valid value range: 0x1~0xFFFF	Sensor scheduling cycle indicating how often the sensor wakes up to work, the real sensor sampling periodicity equals to SCHEDULING_CYCLE *SCHEDULING_GRANULARITY (refer to the general configurations section on how to configure SCHEDULING_GRANULARITY)
N*0x12 + 0x30	SAMPLE_NUMBER	2	Valid value range: 0x1~0xFFFF	Enable or disable dense data sampling for the sensor i.e. the sensor will be sampled for multiple times identified by SAMPLE_NUMBER in one scheduling cycle (which is identified by SCHEDULING_CYCLE *SCHEDULING_GRANULARITY), the time interval between 2 dense data samplings is specified by SAMPLE_CYCLE below. If SAMPLE_NUMBER > 1, then the sensor is enabled with dense data sampling. So far only 2-axis accelerator and 3-axis accelerator sensor types support dense data sampling and reporting, and only one sensor can be enabled with dense data sampling at one time. If SAMPLE_NUMBER = 1, then the sensor is disabled with dense data sampling, i.e. the sensor will be sampled for once in one scheduling cycle
N*0x12 + 0x32	SAMPLE_CYCLE	2	Valid value range: 0x1~0xFFFF	Sample cycle in unit of millisecond for dense data sampling of the sensor. Valid only if SAMPLE_NUMBER > 1 For external proximity sensor, SAMPLE_CYCLE has a special usage as shown below: 1. If SAMPLE_CYCLE = 0, then software will keep counting the sensor event in the whole scheduling cycle defined in SCHEDULING_CYCLE, i.e. in this case, the external proximity sensor is always powered on 2. If SAMPLE_CYCLE > 0, then software will only count the sensor event for the duration specified by this parameter in one scheduling cycle, then power off the sensor in the rest of the scheduling cycle in order to save power
N*0x12 + 0x34	DATA_THRESHOLD1	2	Valid value range: 0x0~0xFFFF	First threshold used to selectively report data. The usage of the high threshold depends on

				CROSS_THRESHOLD_REPORT_CTRL. 1. If CROSS_THRESHOLD_REPORT_CTRL is 0, then DATA_THRESHOLD1 is an unsigned number, and below shows its usage (DATA_THRESHOLD2 is ignored in this case): ▪ If DATA_THRESHOLD1 equals to 0, then the sampled data of the sensor will always be reported, i.e. the feature to report data according to threshold is disabled ▪ Otherwise, only if the ABS value of sampled data is >= the DATA_THRESHOLD1, then the data will be reported 2. If CROSS_THRESHOLD_REPORT_CTRL is 1, both DATA_THRESHOLD1 and DATA_THRESHOLD2 are signed numbers. Below shows the usage of them: ▪ If DATA_THRESHOLD1 < DATA_THRESHOLD2, then only when the sampled data is in between the 2 thresholds (i.e. DATA_THRESHOLD2 >= sampled data >= DATA_THRESHOLD1), the data will be reported. ▪ If DATA_THRESHOLD1 >= DATA_THRESHOLD2, then if the sampled data >= DATA_THRESHOLD1 or the sampled data <= DATA_THRESHOLD2, the data will be reported. The 2 thresholds are sensor specific, the definition and unit are same as that defined in sensor data report message. Special notes: 1. If there are multiple data bundled with one data type (for example tilt, accelerator, or GPS), WxS8000 should report the multiple-data if anyone of the data crosses the threshold 2. For temperature+humidity sensor type, the threshold value is only applied to temperature, so the humidity will be reported always 3. For PIR sensor and external digital sensor, if the threshold is set to 1, then only if the sampled data is 1, the data (1) will be reported
N*0x12 + 0x36	DATA_THRESHOLD2	2	Valid value range: 0x0~0xFFFF	Second threshold used to selectively report data. Applicable only if CROSS_THRESHOLD_REPORT_CTRL is 1, refer to DATA_THRESHOLD1 attribute above for the usage of DATA_THRESHOLD2

WxS8000 supports very powerful edge computing capabilities via combining/configuring different attributes in PPB. Below demonstrates a typical use case setting example. The application scenarios of the example is to use the external proximity sensor to monitor the walkway or escalator running status (via counting the step speed), if no exception, WxS8000 will report the speed counter once a hour, otherwise (the walkway or escalator stops, runs faster or slower), WxS8000 will report the speed counter immediately. To make a good balance between power saving and monitoring performance, the proximity sensor scheduling cycle is set to 5 minutes, and working duration is set to 20 seconds, i.e. the sensor will work/be powered on for 20 seconds every 5 minutes.

Assume in normal case (no exception), the walkway or escalator step will cross the proximity sensor once a second:

1. Set the sensor scheduling cycle as 5 minutes i.e. set SCHEDULING_CYCLE*SCHEDULING_GRANULARITY to 300
2. Set the sensor working duration as 20 seconds, i.e. set SAMPLE_CYCLE to 20000
3. Enable the cross-threshold data report, i.e. set CROSS_THRESHOLD_REPORT_CTRL to 1, set DATA_THRESHOLD1 to 30, DATA_THRESHOLD2 to 10, i.e. if the counter counted in the working duration (20 seconds) is <= 10 or >=30, then this is treated as an abnormal case, and software will report the counter to LoRa server immediately, otherwise, software won't do any report in the whole scheduling cycle
4. Enable global periodic data report cycle as 1 hour, i.e. set PERIODIC_DATA_REPORT_CYCLE* SCHEDULING_CYCLE*SCHEDULING_GRANULARITY to 3600, so that software will always report the counter sampled in current scheduling cycle at the end of global periodic data report cycle regardless whether the counter crosses the threshold or not

4.1.2. Set Individual Attribute in PRB (PlssSetPRB)

Attribute	Description	ValueLength (bytes)	Value	Comment
0x0	CLEAR_PRB	2	0x0000	Clear PRB, so that WxS8000 will use the setting in PPB to communicate with gateway/server in next reboot. As PRB will be updated every 30 minutes, so it is necessary to power off or reboot WxS8000 immediately after clear PRB, otherwise PRB will be filled with valid runtime settings once its update timer expires

4.1.3. Set Mote Local Date-time (PlssSetClk)

LoRa server can use this message to set the local date-time of WxS8000.

Size (bytes)	1	6
FRMPayload	0x15	Date-time to be set to WxS8000, below shows the format: YY-MM-DD-HH-MM-SS YY – Year in hex number

		MM – Month in hex number DD – Day in hex number HH – Hour in hex number MM – Minute in hex number SS – Second in hex number For example, 2017-12-14 20:39:40 should be filled as 0x11-0B-0D-14-27-28

4.1.4. Set Counter Sensor (PlssSetExtCntr)

LoRa server can use this message to set the counter variable of the counter sensor on WxS8000. Usually, the command is used to reset the counter variable

Size (bytes)	1	4
FRMPayload	0x19	An UINT32 counter value used to set WxS8000 local counter variable

4.1.5. Get Individual Attribute in PPB (PlssGetPPB)

The PlssGetPPB message can get all the same attributes as PlssSetPPB (refer to section 4.1.1), except that PlssGetPPB message doesn't have the value parameter in FRMPayload, refer to below for the FRMPayload format of PlssGetPPB command:

Size (bytes)	1	1	1
FRMPayload	0x1A	Attribute	ValueLength

- Attribute: the attribute ID as defined in section 4.1.1
- ValueLength: the length of the attribute value in octets

Once WxS8000 receives the PlssGetPPB message, WxS8000 should report the specified attribute value via PPB attribute report message as defined in section 4.2.4.

NOTE: one PlssGetPPB message can only get one attribute at one time.

4.2. Uplink Message

The uplink messages initiated from WxS8000 are the sensor data report or the response to the downlink messages from LoRa server, they are also unconfirmed.

- The message type (MType) in MAC header must be 0x02 (Unconfirmed Data Up).
- The FPort in the MACPayload is 2
- For little endian or big-endian, the uplink message follows the same rule as downlink message EXCEPT below message, in which the 2-byte/4-byte/8-byte fields are all in big endian:
  - 0x7E – Sensor data report message with fixed data length
- The basic format of uplink message:

Size (bytes)	1	1	1..N
FRMPayload	PLSSPayloadIndicator (0xD7)	MSGType	MSGPayload

- PLSSPayloadIndicator: indicate the FRMPayload is Polysense specific payload
- Below table shows the message type definitions:

MSGType	Description	Comment
0x7E	Sensor data report message with fixed data length	WxS8000 initiates the data report autonomously
0x84	Mote information report message	Response to PlssGetInfo message from LoRa server
0x85	Mote local date-time report message	WxS8000 will report its local date-time to LoRa server once it bootup and registers to server successfully. WxS8000 will report 10 such messages continuously.
0x87	Mote statistics report message	Only supported on R1.2.2.1 and beyond firmware releases Part of the response to PlssGetInfo message from LoRa server
0x88	PPB attribute report message	Response to PlssGetPPB message from LoRa server

4.2.1. Sensor Data Report with Fixed Data Length

The fixed report length format is usually used to transmit multiple sensor type data with short and fixed length in one LoRa frame.

Table below defines the data report format with fixed data length

Field name	Byte offset	Length	Value	Comment
PLSS_PAYLOAD_INDICATOR	0	1 byte	0xd7	Indicate this frame is using Polysense payload format
PLSS_PAYLOAD_TYPE	1	1 byte	0x7e	0x7e - Indicate this frame is using fixed data length format
SENSOR[0].TYPE	2	1 byte		Sensor type. Refer to next table about the sensor type and its value format (length)
SENSOR[0].DATA	3	x bytes		Sensor data. This field length is variable for each type. Refer to next table for details
SENSOR[1].TYPE	3+x	1 byte		Second sensor type
SENSOR[1].DATA	4+x	y bytes		Second sensor data
……				

SENSOR[n].TYPE	X	1 byte		Last sensor type
SENSOR[n].DATA	X+1	z bytes		Last sensor data

Table below defines the sensor data type and its value length.

Sensor Data Type	Value Type	Value Length	Unit	Comment
1	INT16	2 bytes	0.1 Celsius	Temperature data. For example: 01-FF-FF is -0.1 Celsius degree, 01-00-0a is +1 Celsius degree.
2	UINT8	1 byte	%1 Humidity	Humidity data. For example: 02-64 is 100% humidity, 02-32 is 50% humidity
3	INT16[3]	6 bytes	1 mg	3-axis (x/y/z) acceleration data. For example, 03-00-0a-ff-fe-00-64 is x=10mg y=-2mg z=100mg
4	INT16[3]	6 bytes	0.01 degree	Three-dimension (x/y/z) inclination angle data. For example, 04-11-94-ee-6c-46-50 is x=+45® y=-45® z=180®
5	INT16[2]	4 bytes	0.1 mg	2-axis (x/y) acceleration data. For example, 05-00-0a-ff-fe is x=1mg y=-0.2mg
6	UNIT16	2 bytes	1 lm	Light level in lumen
7	UINT16	2 bytes	1 mV	Battery voltage. For example: 07-0d-ac is 3.5V
8	UINT8	2 bytes	0.1 dB	Sound level in dB
9	UINT32+UINT32	8 bytes	0.001 second for latitude and longitude	GPS data, Latitude is in the first UINT32, Longitude is in the second UINT32 MAX longitude: 360*60*60*1000 = 1296000000 = 0x4D3F6400 MAX latitude: 180*60*60*1000 = 648000000 = 0x269FB200 The GPS data is in WGS84 coordinate
10	UNIT16	2 bytes	1 ppm	CO concentration in ppm (part per million)
11	UNIT16	2 bytes	1 mV	External sensor data (analog) in voltage
12	UNIT8	1 byte	NA	External sensor data (digital), 0 or 1
13	UINT32	4 bytes	1 Pa	Atmosphere pressure in Pa
14	INT16	2 bytes	0.1 Celsius	Temperature data from external temperature sensor (for sensor type 16 and 22 in PPB)

15	INT16	2 bytes	0.01 mm	Displacement data from external displacement sensor
16	UNIT8	1 byte	NA	PIR data, 1 – detected, 0 – non-detected
17	UINT8	1 byte	%1 Humidity	Soil humidity from external sensor
18	UINT8	1 byte	NA	PIR data from external PIR sensor 0 –detected 1 – non-detected
19	UINT32	4 bytes	NA	Counter data from external counter sensor The half of total object counter detected in current scheduling cycle, which refers as the total persons (instead of person-time) across the gate.
20	UINT8	1 byte	NA	Switch data from external switch sensor Bit#7: 1 – switch is opened (abnormal status), 0 – switch is closed (normal status) Bit#6~0: the accumulated counter for switch open action, will be restarted from 0 after it reaches 127
21	UNIT16	2 bytes	1 ppm	NH3 concentration in ppm (part per million)
22	UNIT16	2 bytes	1 ppm	AsH3 concentration in ppm (part per million)
23	UNIT16	2 bytes	1 ppm	C6H6 concentration in ppm (part per million)
24	UNIT16	2 bytes	1 ppm	Cl2 concentration in ppm (part per million)
25	UNIT16	2 bytes	1 ppm	H2 concentration in ppm (part per million)
26	UNIT16	2 bytes	1 ppm	H2S concentration in ppm (part per million)
27	UNIT16	2 bytes	1 ppm	HCL concentration in ppm (part per million)
28	UNIT16	2 bytes	1 ppm	HCN concentration in ppm (part per million)
29	UNIT16	2 bytes	1 ppm	HF concentration in ppm (part per million)
30	UNIT16	2 bytes	1 ppm	NO2 concentration in ppm (part per million)
31	UNIT16	2 bytes	1 ppm	O3 concentration in ppm (part per million)
32	UNIT16	2 bytes	1 ppm	PH3 concentration in ppm (part per million)
33	UNIT16	2 bytes	1 ppm	SO2 concentration in ppm (part per million)
34	UNIT16	2 bytes	1 ppm	CH4 concentration in ppm (part per million)
35	UNIT16	2 bytes	1 ppm	C2H2 concentration in ppm (part per million)
36	UNIT16	2 bytes	1 ppm	Gasoline concentration in ppm (part per million)
37	UNIT16	2 bytes	1 ppm	C2H4O3 concentration in ppm (part per million)

38	UINT16	2 bytes	1 mm	Distance
39	UINT16	2 bytes	0.1%	CO2 concentration in %
40	UINT16	2 bytes	1 ppm	CO2 concentration in ppm
41	UINT16	2 bytes	10mA	Current
42	UINT16	2 bytes	1 PSI	Pressure in PSI (Pounds per Square Inch)
43	UNIT16[3]	6 bytes	ug/m3	Air quality/PM for PM1.0, PM2.5 and PM10

4.2.2. Sensor Data Report with Variable Data Length

Table below defines the data report format with variable data length. Such data report usually is used to report a block of data with the same data type but variable data length, and such data block may have to be transmitted in multiple sequential LoRa frames.

Such block of data is called as batch data sampled in dense data sampling in one scheduling cycle. One batch of data is identified by SEQUENCE_ID (bit14~0), it will be increased by one when the next batch of data (sampled in next scheduling cycle) is ready to be transmitted, and wraround to 0 after reaching the MAX value (0x7FFF). SEQUENCE_ID enables LoRa server to distinguish the date sampled in different scheduling cycle, so that server won't put the data from different scheduling cycle together mistakenly.

One batch of data consists of multiple pages (identified by PAGE_ID), one page corresponds to one LoRa frame, and one page can transmit up to 96 x 2 bytes of data. The PAGE_ID enables LoRa server to justify if there is page (LoRa frame) missed during the transmission. If there are pages missed, server should report the first missed page to WxS8000 immediately, then WxS8000 will give up current transmission, and start retransmission from the missed page to the end page of current batch.

The most significant bit (bit15) of SEQUENCE_ID is the transmission end flag of current batch of data, in the last page (LoRa frame), bit15 should be set as 0, otherwise it should be set as 1 (means the transmission isn't completed yet).

Once the batch of data transmission is done, WxS8000 will send another frame (confirmation frame) in which the SEQUENCE_ID is same as the last page, but the PAGE_ID is zero, and there is no sensor data included. The purpose is to request server to confirm if it receives all pages correctly, server should respond with PlssDtRpResp message to inform WxS8000 if it receives all pages correctly or the first missed page (to request WxS8000 to retransmit). If WxS8000 doesn't receive PlssDtRpResp message from server, then it will resend the confirmation frame, and give up after trying for 5 times, then move to next batch of data transmission.

Field name	Byte offset	Length	Value	Comment
PLSS_PAYLOAD_INDICATOR	0	1 bytes	0xD7	Indicate this frame is using Polysense payload format
PLSS_PAYLOAD_TYPE	1	1 bytes		0x7F - Indicate this frame reports 3-axis vibration data with variable data length 0x81 - Indicate this frame reports 2-axis vibration data

				with variable data length 0x82 - Indicate this frame reports sound level data with variable data length
PAGE_ID	2	1 byte	0~255	PAGE_ID from 1 to 255 is used to identify one specific page in one batch of data transmission 0 is used by WxS8000 and LoRa server to confirm if the batch of data transmission is done successfully or not, If PAGE_ID is zero, there is NO DATA[0]..DATA[95] fields in frame anymore
HIGH_SEQUENCE	3	1 byte		Bit7 is the transmission end flag of current batch of data 0 – this is the last page 1 – this is NOT the last page Bit 6~0 is high 7 bits of the SEQUENCE ID
LOW_SEQUENCE	4	1 byte		Low byte of the SEQUENCE_ID, for example if the SEQUENCE_ID is 0x0001, bit 6~0 of HIGH_SEQUENCE should be filled with 0x00, LOW_SEQUENCE should be filled with 0x01
DATA[0]	5	2 bytes		The first data in this frame, also the first or oldest data in this round mass data sampling. For the 3-axis vibration data (PLSS_PAYLOAD_TYPE == 0x7f), this is the X, Y or Z data in unit of 1mg For the 2-axis vibration data (PLSS_PAYLOAD_TYPE == 0x81), this is the X or Y data in unit of 1mg For the sound level data (PLSS_PAYLOAD_TYPE == 0x82), this is the sound level in unit of 0.1dB
DATA[1]	7	2 bytes		The second data in this frame
……				
DATA[95]	195	2 bytes		The last data in this frame

The below example shows how a block of 3-axis vibration data is transmitted through multiple frames using the above variable data format.

3-axis Vibration Sampling Number: 100

3-axis Vibration Sampling Data Block:

INT16 data[300]

Data[0] ….. X data for #0 sampling in unit of 1mg

Data[1] ….. Y data for #0 sampling in unit of 1mg

Data[2] ….. Z data for #0 sampling in unit of 1mg

Data[3] ….. X data for #1 sampling in unit of 1mg

Data[4] ….. Y data for #1 sampling in unit of 1mg

Data[5] ….. Z data for #1 sampling in unit of 1mg

……

Data[297] ….. X data for #99 sampling in unit of 1mg

Data[298] ….. Y data for #99 sampling in unit of 1mg

Data[299] ….. Z data for #99 sampling in unit of 1mg

Frame #0 reported by WxS8000:

D7-7F-01-80-00-Data[0]-Data[1]-Data[2]-…[Data[95]

Frame #1 reported by WxS8000:

D7-7F-02-80-00-Data[96]-Data[97]-Data[98]-…[Data[191]

Frame #2 reported by WxS8000:

D7-7F-03-80-00-Data[192]-Data[193]-Data[194]-…[Data[287]

Frame #3 reported by WxS8000:

D7-7F-04-00-00-Data[288]-Data[289]-Data[290]-…[Data[299]

Frame #4 reported by WxS8000:

D7-7F-00-00-00

Frame responded by LoRa server if all frames are received correctly:

16-00

Frame responded by LoRa server if uplink frame#2 is missed:

16-03

WxS8000 should retransmit Frame#2 to Frame#3, then resend Frame#4 to confirm again.

4.2.3. Mote Information Report

As the response to PlssGetInfo message from LoRa server, WxS8000 should report below mote information to LoRa server in next uplink transmission window after it receives the PlssGetInfo message.

Size (bytes)	4	4	4
MSGPayload	Version	SysTick	(Reserved)

- Version: WxS8000 firmware version, for example, R1.2.3.4 is represented as 0x01020304 in the message

- SysTick: WxS8000 system tick elapsed after last reboot

4.2.4. Mote Local Date-time Report

Every time after WxS8000 bootup and registers to server successfully, it will keep reporting the current local date-time message to server for 10 times. As there is time interval between 2 transmissions, so the date-time in each transmission will be different.

Size (bytes)	1	1	1	1	1	1
MSGPayload	Year	Month	Day	Hour	Minute	Second

The Year, Month, … or Second is saved in Hex number in one byte, for example, 2017-12-14 20:39:40 is saved as 0x11-0B-0D-14-27-28 in the message.

In addition, this report is also part of the response to PlssGetInfo message from LoRa server.

4.2.5. Mote Statistics Report

As part of the response to PlssGetInfo message from LoRa server, WxS8000 should report mote statistics report as well as other reports to LoRa server in next uplink transmission window after it receives the PlssGetInfo message.

Size (bytes)	2	2	2	2	2	2	2	2	2	2	1
MSGPayload	TxErr	Rx2	RxMcst	RxErr	RxCrcErr	RxPtErr	RxPlErr	RxAddrErr	RxUnsup	Rssi	Snr

- TxErr: transmit frame errors on MAC layer
- Rx2: received frames on Rx2 slot
- RxMcst: received multicast frames
- RxErr: received error frames
- RxCrcErr: received frames with wrong CRC
- RxPtErr: received frames with wrong or unexpected protocol
- RxPlErr: received frames with wrong payload or payload length
- RxAddrErr: received frames with wrong device address
- RxUnsup: received frames with unsupported MType
- Rssi: downlink RSSI measured by WxS8000
- Snr: downlink SNR measured by WxS8000

4.2.6. PPB Attribute Report

PPB attribute report is the response to PlssGetPPB message, below shows the message format:

Size (bytes)	1	1..n	
MSGPayload	ValueLength	Value	

- ValueLength: indicate the length of the value in octets

- Value: the value of the PPB attribute specified in PlssGetPPB message

5. Firmware Upgrade

WxS8000 supports firmware upgrade via ST-Link being connected to its local JTAG interface, and supports OTA firmware upgrade from LoRa server.

Note: starting from R2.3.x, WxS8000 introduces 2 images (Loader + Application) in order to support OTA firmware upgrade, the first release supporting OTA firmware upgrade is Loader - R1.0.1 and Application – R2.3.1. Before that, WxS8000 only supports single image and the release version is R2.2.x.

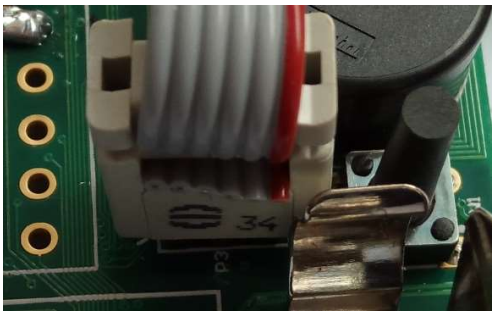
5.1. Firmware Upgrade via ST-Link

This section describes how to do the firmware upgrade via the JTAG interface of WxS8000. As a generic way, this approach can upgrade old WxS8000 firmware (single image) and new WxS8000 firmware (Loader + Application images).

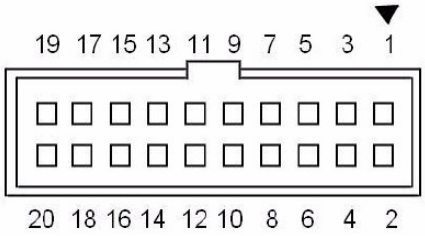
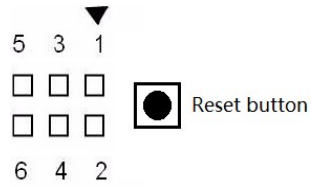
5.1.1. Setup ST-LINK and ST-LINK Utility

Before to access PPB, you must have below stuffs available:

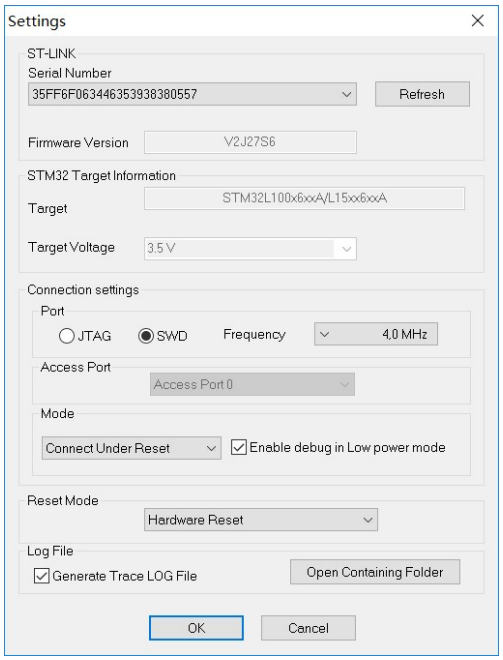
1. The ST-LINK device
2. A USB cable
3. The special JTAG cable to connect the ST-LINK to the JTAG interface of WxS8000
4. Download STM32 ST-LINK Utility from linkage http://www.st.com/content/st_com/en/products/embedded-software/development-tool-software/stsw-link004.html#getsoftware-scroll, and install STM32 ST-LINK Utility on PC
   - Unzip the compressed file
   - Run STM32 ST-LINK Utility v4.0.0 setup.exe directly
5. Download STM32 ST-LINK USB driver from linkage http://www.st.com/content/st_com/en/products/embedded-software/development-tool-software/stsw-link009.html, and install on PC
   - Unzip the compressed file
   - Run dpinst_x86.exe for 32-bit Windows system or dpinst_amd64.exe for 64-bit Windows system
6. Connect the ST-LINK to PC via the USB cable
7. Connect the ST-LINK to WxS8000 via the JTAG cable
   - The big plug of the JTAG cable should be plugged into ST-LINK 20-pin JTAG connector
   - The small plug of the JTAG cable should be plugged into WxS8000 JTAG connector (the red cycle in Diagram 2-2). After being plugged in, the red wire of the cable should be aligned with pin#2 of the WxS8000 JTAG connector (i.e. towards the reset button), refer to diagram below



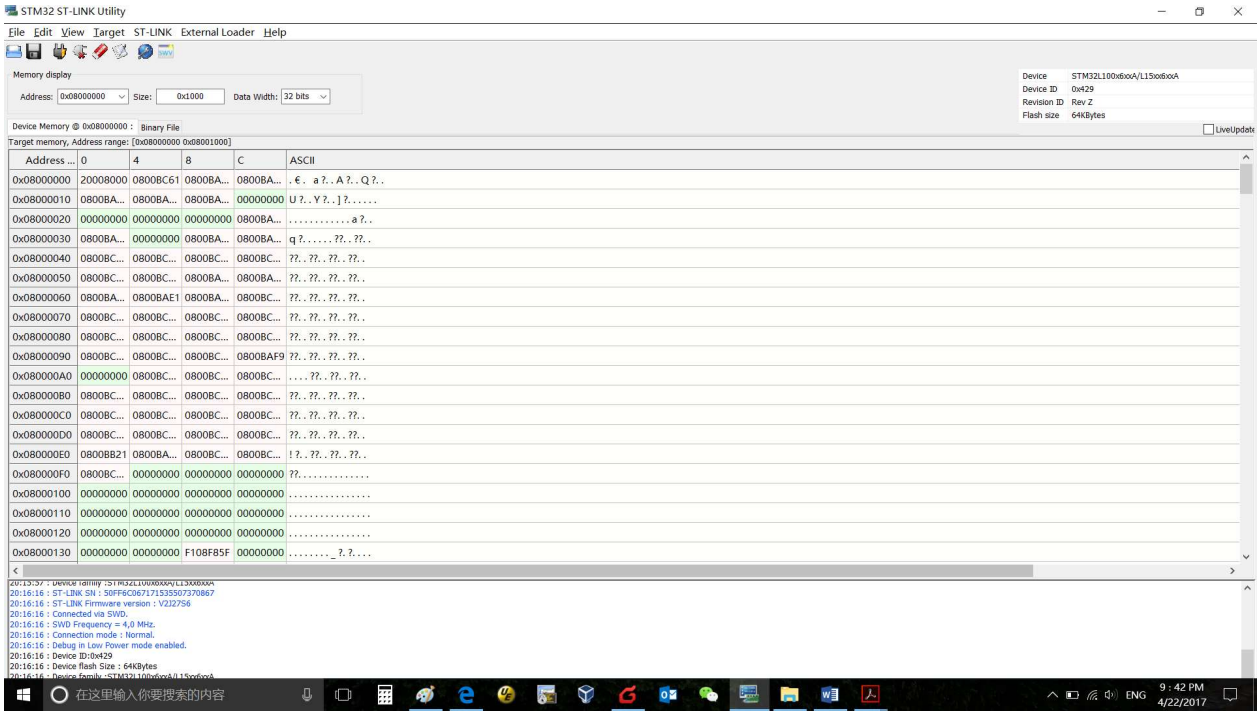
NOTE: If you don't have the special JTAG cable, you can make direct connection between ST-LINK JTAG pins and WxS8000 JTAG pins according to the mappings in below table:

ST-LINK JTAG pin layout	ST-LINK JTAG pin#	WxS8000 JTAG pin#	WxS8000 JTAG pin layout
	1	2	
	7	4	
	9	3	
	15	5	
	20	6	

8. Start STM32 ST-LINK Utility on PC, and make sure to follow below settings (clicking [Target] -> [Settings] menu item to open the settings dialog)



9. Click [Target] -> [Connect] menu item, then diagram below 4.1-1 will show up once the connection is done successfully

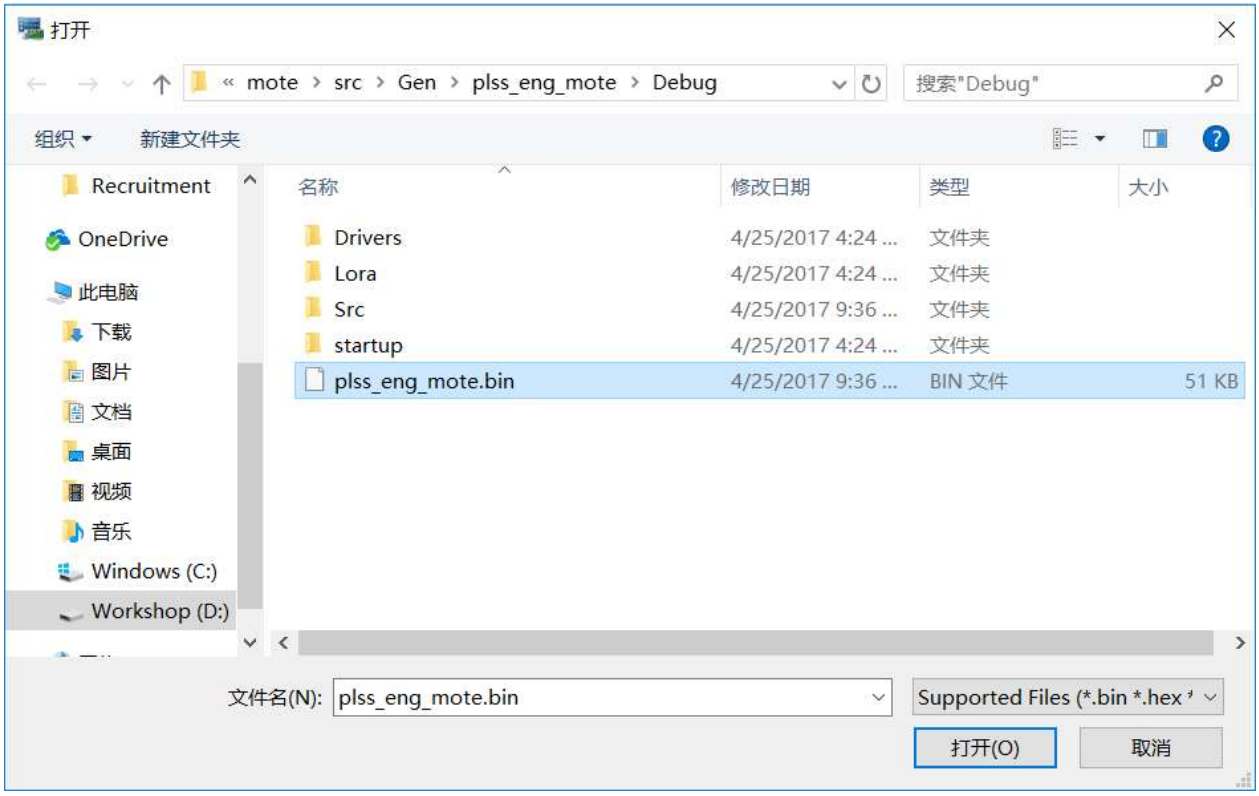


5.1.2. Firmware Upgrade

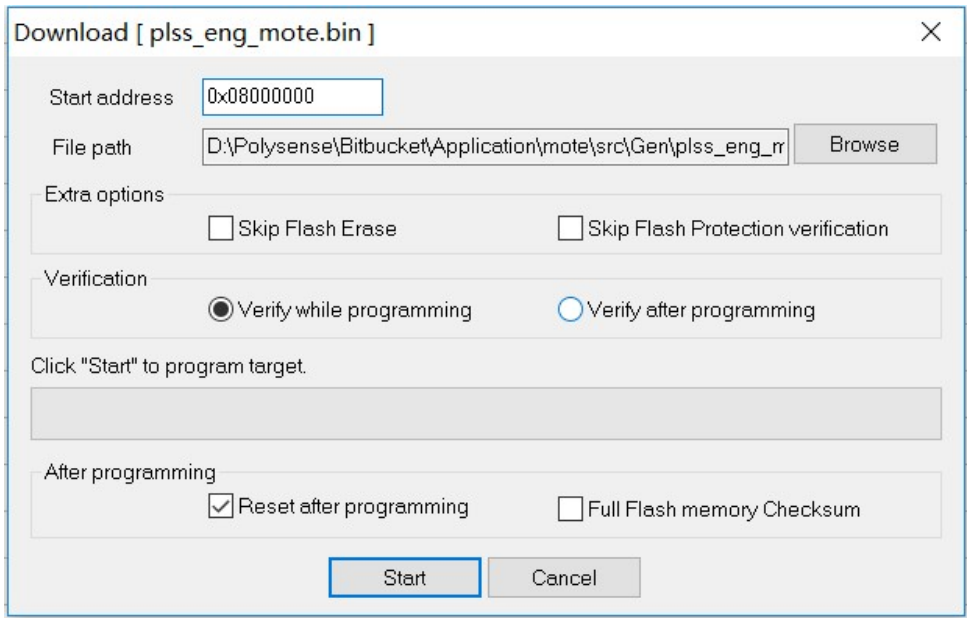
Once the ST-Link and ST-Link Utility are set up properly, user can follow below procedure to do the firmware upgrade.

5.1.2.1. Upgrade Old WxS8000 Firmware – Single Image

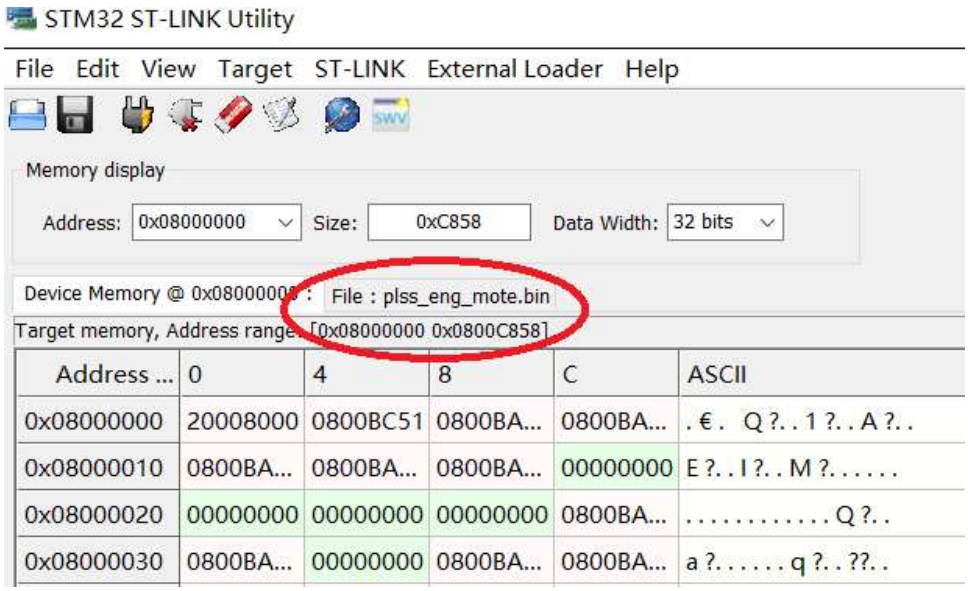
1. lick [Target] -> [Program&Verify] menu item in ST-Link Utility, and select the new firmware (for example, plss_eng_mote.bin) to be downloaded in the pop-up dialog as shown below, then click [Open] button



2. The below download diagram will be popped up once the [Open] button is clicked, in this diagram, it is important to input 0x08000000 as the start address, which the new firmware will be programmed on flash, and select [Reset after programming] checkbox (so that WxS8000 will be reset automatically after the new firmware downloading), then click [Start] button to start firmware downloading



3. Once the firmware downloading is successful, the new firmware file name and address range will show up on top-left of the STM32 ST-LINK Utility window, refer to diagram below



5.1.2.2. Upgrade New WxS8000 Firmware – Loader + Application Images

Starting from R2.3.x, WxS8000 introduces 2 images (Loader + Application), both can be upgraded via ST-Link by following the same procedure described in 5.1.2.1, except:

1. To upgrade the loader image, the loader image must be select in step#1 of 5.1.2.1 at first, and it need input 0x08000000 as the start address in step#2 of 5.1.2.1

2. To upgrade the application image, the application image must be selected in step#1 of 5.1.2.1 at first, and it need input 0x08008000 as the start address in step#2 of 5.1.2.1

5.2. Firmware Upgrade via OTA

In order to support firmware upgrade via OTA:

1. WxS8000 must have been programmed with Polysense iEdge Loader image R1.0.1 (or later release) and Polysense iEdge Application image R2.3.1 (or later release)
2. The LoRa server must support Polysense proprietary message for firmware upgrade (refer to chapter 4 for the details)

5.2.1. Prepare FW Image

If you are using Polysense LoRa Server, you can login CLI and prepare/load the FW by the server:

1. Copy the new loader images into /shared/OfficialBuild/iEdge/Loader/Lx.x.x folder
   - Lx.x.x identified the loader image version to be upgraded
   - Polysense application image is generic and applied to all kinds of Polysense WxS8xxx motes, but the loader image is mote type and mote hardware revision specific, e.g. WxS8000 and WxS8700 have different loader image, WxS8000 Rev1.0 and WxS8000 Rev1.1 also have different loader image though their loader version is same
   - It is necessary to copy all kinds of loader images with the same loader version into above folder
2. Copy the new application image into /shared/OfficialBuild/iEdge/Rx.x.x folder
   - Rx.x.x identified the application image version to be upgraded
3. Login CLI and enter the “config” node
4. Load loader image with the specific version, for example:

   fwload loader 1.0.1
5. Load application image with the specific version, for example:

   fwload app 2.3.1

5.2.2. Upgrade FW

If you are using Polysense LoRa Server, you can login CLI and start the FW upgrading:

1. Login CLI
2. Navigate to the corresponding mote node which is going to be upgraded with the images loaded above
3. You can choose what kinds of upgrading:

   “mgmt upgrade_all” - Upgrade both loader and application images

   Or

   “mgmt upgrade_app” - Upgrade the application image only

4. Wait for about tens of minutes until the upgrading is done

6. Compliances

FCC Compliance：

This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

Any Changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

Reorient or relocate the receiving antenna.

Increase the separation between the equipment and receiver.

Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.

Consult the dealer or an experienced radio/TV technician for help.

Please take attention that changes or modification not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

FCC RF EXPOSURE STATEMENT:

This equipment should be installed and operated with minimum distance 20cm between the radiator & your body.

ISEDC Compliance

This device complies with Innovation, Science, and Economic Development 7 Canada licence-exempt RSS standard(s). Operation is subject to the following two conditions: (1) This device may not cause in terference, and (2) This device must accept any interference, including interference that may cause undesired operation of the device.

Le présent appareil est conforme aux CNR d' Innovation, Science et Développement économique Canada applicables aux appareils radio exempts de licence. L'exploitation est autorisée aux deux conditions suivantes: (1) l'appareil ne doit pas produire de brouillage, et (2) l'utilisateur de l'appareil doit accepter tout brouillage radioélectrique subi, même si le brouillage est susceptible d'en compromettre le fonctionnement.

This transmitter must not be co-located or operating in conjunction with any other antenna or transmitter. This equipment should be installed and operated with a minimum distance of 20 centimeters between the radiator and your body.

Cet émetteur ne doit pas tre Co-placé ou ne fonctionnant en même temps qu'aucune autre antenne ou émetteur. Cet équipement devrait être installé et actionné avec une distance minimum de 20 centimètres entre le radiateur et votre corps.

Class B Notice for Canada This Class B digital apparatus complies with Canadian ICES-003.

Cet appareil numérique de la classe B est conforme à la norme NMB-003 du Canada.

Contact Information

Additional Polysense assistance and product information can be obtained by contacting Polysense directly at the following locations or by going to the Polysense home page: www.polysense.net

Polysense (Beijing) Technologies Inc.

Address: 26 Shangdi Xinxi Road, Suite 820, Haidian District, Beijing, China 100085

E-mail: support@polysense.net

Polysense (USA) Technologies Inc.

Address: 3000 Scott Blvd, Suite 108, Santa Clara, CA 95054

E-mail: support@polysense.net